# Configuration Manual

MSc Research Project<br>Cloud Computing

Vinay Sriram Iyer
Student ID: X23203595

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Vinay Sriram Iyer |
| **Student ID:** | X23203595 |
| **Programme:** | Cloud Computing |
| **Year:** | 2024 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Yasantha Samarawickrama |
| **Submission Due Date:** | 12/12/2024 |
| **Project Title:** | Configuration Manual |
| **Word Count:** | 4867 |
| **Page Count:** | 19 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | Vinay Sriram Iyer |
| **Date:** | 12th December 2024 |

## PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | □ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# AI Acknowledgement Supplement

## [MSc Research Project]

## [Configuration Manual]

| Your Name/Student Number | Course | Date |
|---|---|---|
| **Vinay Sriram Iyer/x23203595** | MSc in Cloud Computing | 12/12/2024 |

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

## AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

| Tool Name | Brief Description | Link to tool |
|---|---|---|
| **N/A** | N/A | N/A |
| | | |

## Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

| **[N/A]** | |
|---|---|
| [N/A] | |
| [N/A] | [N/A] |

## Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

### Additional Evidence:

[N/A]

### Additional Evidence:

[N/A]

# Configuration Manual

Vinay Sriram Iyer
X23203595

## 1 Preprocessing GWA-T-12 Bitbrains

### 1.1 Tools and Technologies

- **Operating System - Windows 11 Pro**
- **Processor - Intel(R) Core i5**
- **Physical Memory - 16 GB RAM**
- **Platform - Eclipse IDE 2024-03 (4.31.0)**
- **Language - Python 3.12.2**
- **Package - Pip 24.2, Pandas 2.2.3, numpy 2.1.1, python-dateutil 2.9.0.post(), pytz 2024.2, six 1.16.0, tzdata 2024.1, git-filter-repo 2.45.0**

### 1.2 Environmental Setup:

1. Download and install Eclipse IDE from the link here. Download and install Python 3.12.2 from the link here. Ensure the "Add Python to Path" box is checked before clicking install. Ensure the full path and Scripts folders are added to the Python installation directory and pip directory by the paths below:

```
C:\Users\<YourUsername>\AppData\Local\Programs\Python\Python39\
C:\Users\<YourUsername>\AppData\Local\Programs\Python\Python39\Scripts\
```

2. Go to the official Python website and download the get-pip.py script from the URL below:

https://bootstrap.pypa.io/get-pip.py

Save the file as get-pip.py to your home directory, preferably in your Downloads folder. Navigate to this directory in Command Prompt and run the below command to install pip 24.2:

```
C:\Users\<username>\AppData\Local\Programs\Python\Python312\python.exe
get-pip.py
```

3. Verify the installation of Python 3.12.2 and Pip 24.2 by typing the command below in Command Prompt:

```
python --version
pip --version
```

4. Install the Pydev Plugin from Eclipse Marketplace. Once installed, restart Eclipse. Configure Python 3.12.2 and select the location of the Python Interpreter installation. It is recommended to use your home directory on your machine for the Python 3.12.2 Interpreter installation. Open the terminal on Eclipse IDE and install pandas with numpy by the following commands below in the project directory:

```
pip install pandas
pip install numpy
```

Close Eclipse IDE. Reboot the machine and launch the Eclipse IDE once again.

## 1.3 Preprocessing the Dataset:

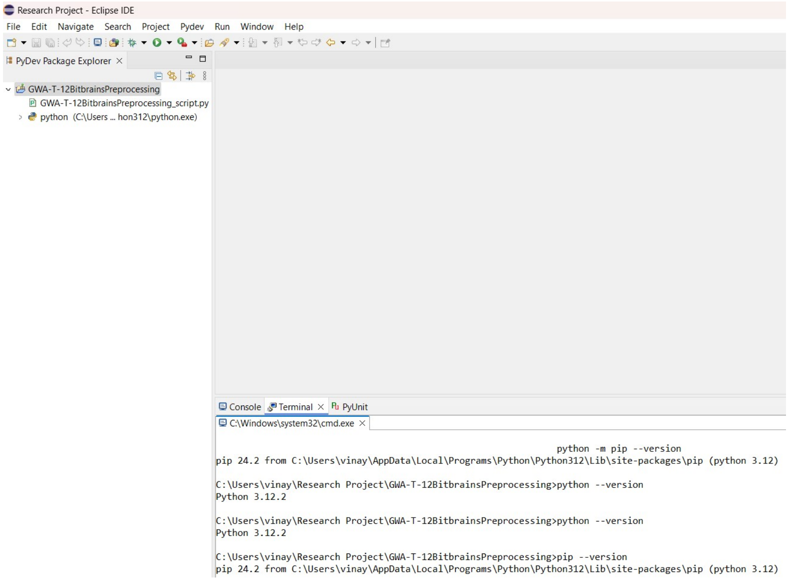
1. Click on File >New >Python PyDev Project. Set the Python Interpreter as 3.12.2. In this configuration, the Pydev Project shall be titled 'GWA-T-12BitbrainsPreprocessing'. Ensure the Python 3.12.2 installation points correctly in the interpreter settings.
2. Navigate to File Explorer and locate the PyDev Module script(.py) that needs to be imported. In this case, the script shall be titled 'GWA-T-12BitbrainsPreprocessing_script.py'. After dropping the script into your project, right-click on 'GWA-T-12BitbrainsPreprocessing' and refresh. Ensure 'GWA-T-12BitbrainsPreprocessing_script.py' is placed in the 'GWA-T-12BitbrainsPreprocessing' Source Folder. Open the terminal in the navigated directory and import 'GWA-T-12BitbrainsPreprocessing_script.py' by the command below:

```
import GWA-T-12BitbrainsPreprocessing_script.py
```

3. Replace the absolute paths of the variables 'fastStorage_dir' and 'rnd_dirs' with the path specific to the username on your machine:

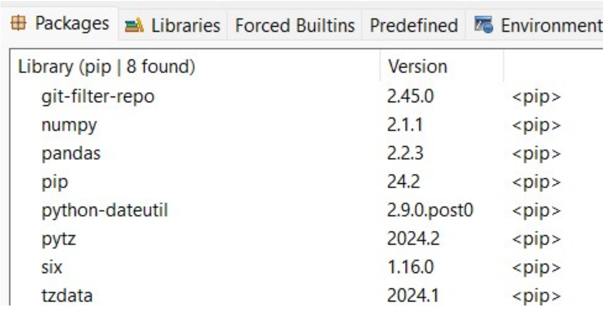
```
C:\Users\<username>\<Folder>\fastStorage\2013-8
C:\Users\<username>\<Folder>\rnd\2013-7
C:\Users\<username>\<Folder>\rnd\2013-8
C:\Users\<username>\<Folder>\rnd\2013-9
```

4. After aggregating the data, the preprocessed results are saved as CSV files (preprocessed_fastStorage.csv and preprocessed_rnd.csv) in 'GWA-T-12BitbrainsPreprocessing'.


Research Project - Eclipse IDE
File Edit Navigate Search Project Pydev Run Window Help
PyDev Package Explorer
GWA-T-12BitbrainsPreprocessing
GWA-T-12BitbrainsPreprocessing_script.py
python (C:\Users ... hon312\python.exe)
Console Terminal PyUnit
C:\Windows\system32\cmd.exe

```
                                                               python -m pip --version
pip 24.2 from C:\Users\vinay\AppData\Local\Programs\Python\Python312\Lib\site-packages\pip (python 3.12)

C:\Users\vinay\Research Project\GWA-T-12BitbrainsPreprocessing>python --version
Python 3.12.2

C:\Users\vinay\Research Project\GWA-T-12BitbrainsPreprocessing>python --version
Python 3.12.2

C:\Users\vinay\Research Project\GWA-T-12BitbrainsPreprocessing>pip --version
pip 24.2 from C:\Users\vinay\AppData\Local\Programs\Python\Python312\Lib\site-packages\pip (python 3.12)
```

Figure 1: Final Environmental Setup for preprocessing the GWA-T-12 Bitbrains dataset on Eclipse IDE



| Library (pip \| 8 found) | Version | |
|---|---|---|
| git-filter-repo | 2.45.0 | <pip> |
| numpy | 2.1.1 | <pip> |
| pandas | 2.2.3 | <pip> |
| pip | 24.2 | <pip> |
| python-dateutil | 2.9.0.post0 | <pip> |
| pytz | 2024.2 | <pip> |
| six | 1.16.0 | <pip> |
| tzdata | 2024.1 | <pip> |

Figure 2: Libraries for the 'GWA-T-12BitbrainsPreprocessing' Project Folder on Eclipse IDE

# 2 Incorporated Aggregated Algorithmic Programs

## 2.1 Tools and Technologies

- **Operating System - Windows 11 Pro**
- **Processor - Intel(R) Core i5**
- **Physical Memory - 16 GB RAM**
- **Platform - Eclipse IDE 2024-03 (4.31.0)**
- **Language - Java (jdk-17)**
- **Framework - CloudSim-3.0.3 Toolkit**

## 2.2 Environmental Setup

1. Download and install CloudSim jar files with all related dependencies from the link here. Extract the files to a folder on your local machine, preferably the Downloads folder.
2. Create a new Eclipse IDE Workspace Directory. For this configuration, the Workspace Directory shall be titled 'Cloudsim_ResearchProject'. Open 'Cloudsim_ResearchProject' and go to '`File > New > Java Project`. For the configuration of this Project Directory, the directory shall be titled as 'cloudsimresearchprojectfinal'. Check the box that says "Use default location" to point to 'cloudsimresearchprojectfinal'.
3. Right click on 'cloudsimresearchprojectfinal' and go to 'Properties'. Go to `'Java Build Path' > 'Libraries'`. Click on 'Add External JARs'. Select all the .jar files in the extracted CloudSim jar folder from the Downloads folder. Click on 'Apply and Close'.
4. Click on 'src' inside the project folder 'cloudsimresearchprojectfinal' and click on 'Remove from Build Path'. Delete the 'src folder'. Right Click on 'cloudsimresearchprojectfinal' and select `'New' > 'Source Folder'`. Create java/src/main as the new java source folder for 'cloudsimresearchprojectfinal'. Navigate to File Explorer and locate the Java classes that needs to be imported in 'cloudsimresearchprojectfinal'. For this configuration, the aggregated algorithmic programs shall be utilised for evaluating scalability on CloudSim-3.0.3 in this section and AWS in the next section. It is recommended to keep the aggregated algorithmic program classes in a folder on your machine, preferably the Downloads Folder. In this configuration, the folder containing the aggregated algorithmic program classes shall be titled 'cloudsimresearchprojectfinalnew'. Drag and drop the aggregated algorithmic program classes titled 'AggregatedIterativeHeuristicNonConvexEnergyAware', 'AggregatedHeuristicAugment' and 'AggregatedPrioritySelectionOffloading'.
5. The aggregated algorithmic programs that shall be implemented on CloudSim-3.0.3 and on AWS us-east-1(North Virginia) Region in the next section will incorporate the last iteration on simulation. For each aggregated algorithmic program, change the absolute file path for the simulation output according to the lines below. Note that the backward slashes `'\\'` shall be represented as forward slashes `'//'` here:

**Iterative Heuristic Energy-Aware Non-Convex Algorithmic Program:**
PrintStream out = new PrintStream(new FileOutputStream( "C://Users//<`username` >//Cloudsim_ResearchProject//
simulation_output_aggregatediterativeheuristicnonconvexenergyawareupdatednewtwo.txt"));

writeCloudletResultsToFile(newList, "C://Users//<`username` >//
Cloudsim_ResearchProject
simulation_output_aggregatediterativeheuristicnonconvexenergyawareupdatednewtwo.txt");

cloudletList = loadCloudletsFromCSV(brokerId, "C:/Users/<`username` >/Research Project/GWA-T-12BitbrainsPreprocessing/preprocessed_fastStorage.csv");

cloudletList.addAll(loadCloudletsFromCSV(brokerId, "C:/Users/<`username` >/Research Project/GWA-T-12BitbrainsPreprocessing/preprocessed_rnd.csv"));

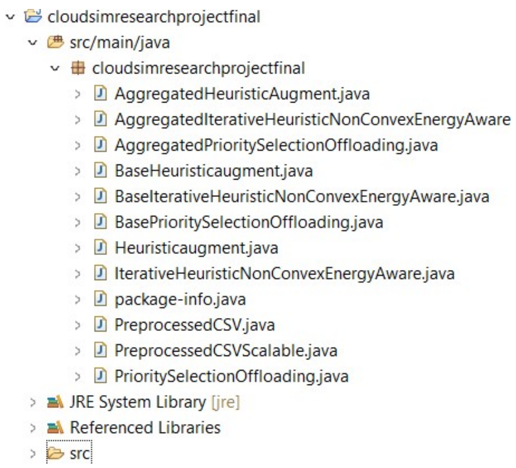
**Heuristic AUGMENT Non-Convex Algorithmic Program:**

```
PrintStream out = new PrintStream(new FileOutputStream(”C://Users//<username
>//Cloudsim_ResearchProject
//simulation_output_aggregatedheuristicaugmentupdatednewthree.txt”));

cloudletList = loadCloudletsFromCSV(brokerId, ”C:/Users/<username >/Research
Project/GWA-T-12BitbrainsPreprocessing/preprocessed_fastStorage.csv”);

cloudletList.addAll(loadCloudletsFromCSV(brokerId, ”C:/Users/<username >/Research
Project/GWA-T-12BitbrainsPreprocessing/preprocessed_rnd.csv”));

writeEnergyResultsToFile(”C://Users//<username >//Cloudsim_ResearchProject
//simulation_output_aggregatedheuristicaugmentupdatednewthree.txt”, totalEnergy, cool-
ingPower, totalEnergyWithCooling);

writeCloudletResultsToFile(newList, ”C://Users//<username>//
Cloudsim_ResearchProject
//simulation_output_aggregatedheuristicaugmentupdatednewthree.txt”);
```

**Priority Selection Offloading Algorithmic Program:**

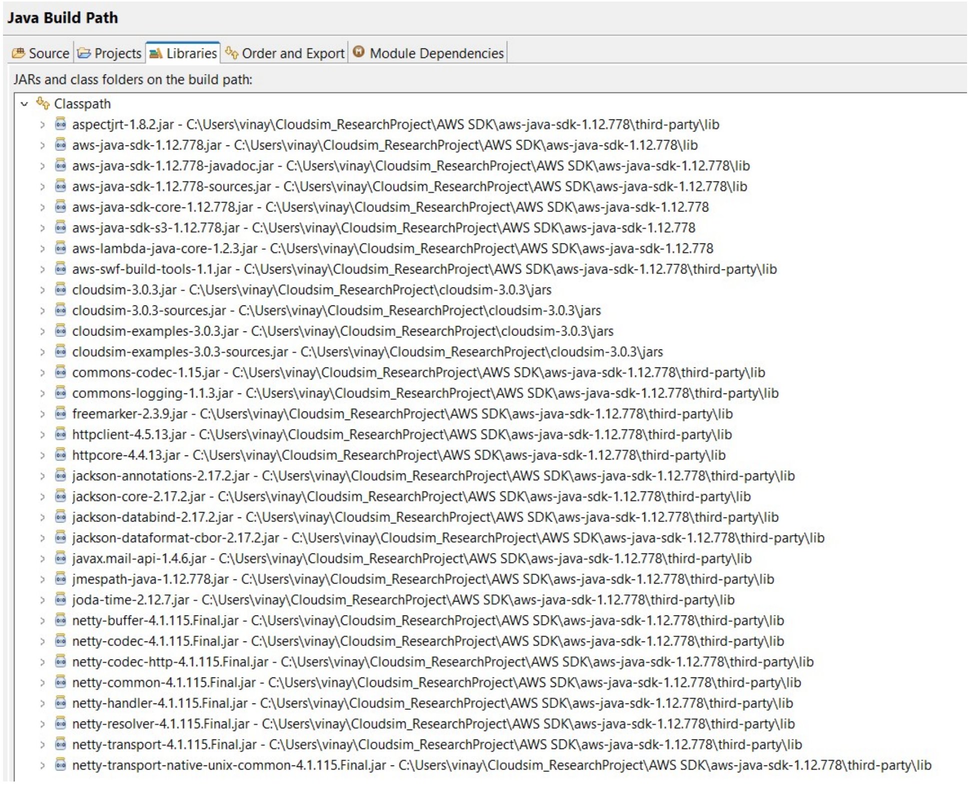
```
String outputFilePath = ”C://Users//<username>//Cloudsim_ResearchProject
//simulation_output_aggregatedpriorityselectionoffloadingalgorithmnewfive.txt”;

cloudletList = loadCloudletsFromCSV(brokerId,”C:/Users/<username>/Research
Project/GWA-T 12BitbrainsPreprocessing/preprocessed_fastStorage.csv”);

cloudletList.addAll(loadCloudletsFromCSV(brokerId, ”C:/Users/<username>/Research
Project/GWA-T-12BitbrainsPreprocessing/preprocessed_rnd.csv”));
```

6. Run each algorithmic program for the CloudSim simulation. CloudSim will execute each simulation and the simulation output file will be stored in the respective directories for each algorithmic program.



```
cloudsimresearchprojectfinal
  src/main/java
    cloudsimresearchprojectfinal
      AggregatedHeuristicAugment.java
      AggregatedIterativeHeuristicNonConvexEnergyAware
      AggregatedPrioritySelectionOffloading.java
      BaseHeuristicaugment.java
      BaseIterativeHeuristicNonConvexEnergyAware.java
      BasePrioritySelectionOffloading.java
      Heuristicaugment.java
      IterativeHeuristicNonConvexEnergyAware.java
      package-info.java
      PreprocessedCSV.java
      PreprocessedCSVScalable.java
      PrioritySelectionOffloading.java
  JRE System Library [jre]
  Referenced Libraries
  src
```

Figure 3: Final Environmental Setup for the Incorporated Aggregated Algorithmic Programs on Eclipse IDE.



JARs and class folders on the build path:

```
Classpath
  aspectjrt-1.8.2.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  aws-java-sdk-1.12.778.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\lib
  aws-java-sdk-1.12.778-javadoc.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\lib
  aws-java-sdk-1.12.778-sources.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\lib
  aws-java-sdk-core-1.12.778.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778
  aws-java-sdk-s3-1.12.778.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778
  aws-lambda-java-core-1.2.3.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778
  aws-swf-build-tools-1.1.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  cloudsim-3.0.3.jar - C:\Users\vinay\Cloudsim_ResearchProject\cloudsim-3.0.3\jars
  cloudsim-3.0.3-sources.jar - C:\Users\vinay\Cloudsim_ResearchProject\cloudsim-3.0.3\jars
  cloudsim-examples-3.0.3.jar - C:\Users\vinay\Cloudsim_ResearchProject\cloudsim-3.0.3\jars
  cloudsim-examples-3.0.3-sources.jar - C:\Users\vinay\Cloudsim_ResearchProject\cloudsim-3.0.3\jars
  commons-codec-1.15.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  commons-logging-1.1.3.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  freemarker-2.3.9.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  httpclient-4.5.13.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  httpcore-4.4.13.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  jackson-annotations-2.17.2.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  jackson-core-2.17.2.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  jackson-databind-2.17.2.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  jackson-dataformat-cbor-2.17.2.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  javax.mail-api-1.4.6.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  jmespath-java-1.12.778.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  joda-time-2.12.7.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-buffer-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-codec-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-codec-http-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-common-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-handler-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-resolver-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-transport-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
  netty-transport-native-unix-common-4.1.115.Final.jar - C:\Users\vinay\Cloudsim_ResearchProject\AWS SDK\aws-java-sdk-1.12.778\third-party\lib
```

Figure 4: Libraries for the 'cloudsimresearchprojectfinal' Project Folder on Eclipse IDE. The AWS SDK library is included here as well which shall be described and utilised in the next section.

# 3 AWS Testing

## 3.1 Tools and Technologies

- **Region - us-east-1 (N. Virginia)**

- **Platform - Eclipse IDE 2024-03 (4.31.0), AWS**

- **AWS Services - Key Management Service us-east-1-KMS-Requests, Elastic Compute Cloud T3CPU, EBS General Purpose (gp3), Elastic Compute Cloud NatGateway, Internet Gateway, Virtual Private Cloud EC2 Free Tier, Virtual Private Cloud VpcEndpoint, Virtual Private Cloud VPCPeering, CloudWatch, Lambda, Simple Storage Service GeneralPurposeBuckets**

- **Language - Java (jdk-17)**

- **Framework - CloudSim-3.0.3 Toolkit**

- **Libraries - AWS S3 SDK 2.20.16, AWS Lambda SDK 2.20.16, AWS Lambda Java Core 1.2.1, AWS S3 (Legacy SDK) 1.12.507, Amazon CloudWatch SDK 2.20.16, AWS SDK Core 2.20.16, CloudSim 3.0.3, SLF4J API 2.0.9, SLF4J Simple 2.0.9, Maven Shade 3.4.0, Maven Compiler 3.10.1**

## 3.2 Environmental Setup

1. Sign in to the AWS Management Console and navigate to the 'VPC dashboard'. Click on 'Create VPC'. For this configurational setup, the first VPC shall be titled 'NetworkSlicing1_VPC'. Specify the IPv4 CIDR block as 10.1.0.0/16 and select tenancy as default. Repeat this step for four more VPC's and accordingly name them as 'NetworkSlicing2_VPC', 'NetworkSlicing3_VPC', 'NetworkSlicing4_VPC' and 'NetworkSlicing5_VPC'. Accordingly assign the IPv4 CIDR blocks respectively as 10.2.0.0/16, 10.3.0.0/16, 10.4.0.0/16, 10.5.0.0/16. Click on 'Edit VPC settings' for each VPC and ensure 'Enable DNS resolution' and 'Enable DNS hostnames' are enabled and saved for each VPC.
2. Navigate to the 'Route Table' dashboard. Click on 'create route table'. For this configurational setup, the first route table shall be titled 'RouteTable1_Subnet'. Select 'NetworkSlicing1_VPC' and create the route table. Repeat this step for 'NetworkSlicing2_VPC', 'NetworkSlicing3_VPC', 'NetworkSlicing4_VPC' and 'NetworkSlicing5_VPC' for 'RouteTable2_Subnet', 'RouteTable3_Subnet', 'RouteTable4_Subnet' and 'RouteTable5_Subnet'.
3. Navigate to the 'Subnets' dashboard. Click on 'Create subnet'. Select 'NetworkSlicing1_VPC'. For this configurational setup, the the subnet shall be titled as 'NetworkSlice1_Subnet'. Assign IPv4 CIDR block as 10.1.1.0/24 and create the subnet. Repeat this step for for four more subnets and accordingly name them as 'NetworkSlice2_Subnet', 'NetworkSlice3_Subnet', 'NetworkSlice4_Subnet', 'NetworkSlice5_Subnet'. Accordingly assign the IPv4 CIDR blocks respectively as 10.2.1.0/24, 10.3.1.0/24, 10.4.1.0/24 and 10.5.1.0/24.
4. Go back to the 'Route Table' dashboard. Click on 'RouteTable1_Subnet'. Go to 'Subnet associations' > 'Add subnet associations'. Select 'NetworkSlice1_Subnet' and click on 'Save associations'.
Repeat this step for 'RouteTable2_Subnet', 'RouteTable3_Subnet', 'RouteTable4_Subnet' and 'RouteTable5_Subnet'.
5. Go to the 'Security Groups' dashboard. Click on 'Create Security Group'. For this configurational setup, the security group shall be titled as 'RingFence1_SecurityGroup'. Select 'NetworkSlicing1_VPC' and create the security group. Go to 'Inbound Rules' and click on 'Edit inbound rules'. Add an inbound rule for for 'Type' HTTPS and select 'RingFence1_SecurityGroup' as the source. Click on 'Save Rules'. Repeat step 5 for 'RingFence2_SecurityGroup' with 'RingFence2_SecurityGroup' as an inbound rule.
Repeat step 5 for nine more security groups till 'RingFence11_SecurityGroup'. For 'RingFence2_SecurityGroup' to 'RingFence11_SecurityGroup', do not add an inbound rule for the same security group.
6. Go to the Instances Dashboard. Click on 'Launch instance'. For this configurational setup, the first instance shall be titled 'NetworkResource0_VM'. Select the 'Instance Type' as 't2.nano'. Select 'NetworkSlicing1_VPC', 'NetworkSlice1_Subnet' and 'RingFence1_SecurityGroup' as the VPC, subnet and security group respectively. Create the first instance. Create 29 more instances with 'Instance Type' set as 't2.nano' till 'NetworkResource29_VM' with the following configuration as per the table below:

Table 1: Configuration of Network Resources, VPC's, Subnets, and Security Groups

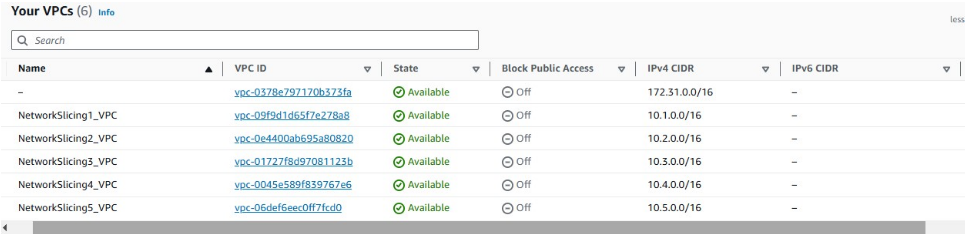
| Network Resource (VM) | VPC | Subnet | Security Group |
|---|---|---|---|
| NetworkResource1_VM | NetworkSlicing1_VPC | NetworkSlice1_Subnet | RingFence1_SecurityGroup |
| NetworkResource2_VM | NetworkSlicing1_VPC | NetworkSlice1_Subnet | RingFence1_SecurityGroup |
| NetworkResource3_VM | NetworkSlicing1_VPC | NetworkSlice1_Subnet | RingFence2_SecurityGroup |
| NetworkResource4_VM | NetworkSlicing1_VPC | NetworkSlice1_Subnet | RingFence2_SecurityGroup |
| NetworkResource5_VM | NetworkSlicing2_VPC | NetworkSlice2_Subnet | RingFence3_SecurityGroup |
| NetworkResource6_VM | NetworkSlicing2_VPC | NetworkSlice2_Subnet | RingFence3_SecurityGroup |
| NetworkResource7_VM | NetworkSlicing2_VPC | NetworkSlice2_Subnet | RingFence4_SecurityGroup |
| NetworkResource8_VM | NetworkSlicing2_VPC | NetworkSlice2_Subnet | RingFence4_SecurityGroup |
| NetworkResource9_VM | NetworkSlicing2_VPC | NetworkSlice2_Subnet | RingFence4_SecurityGroup |
| NetworkResource10_VM | NetworkSlicing2_VPC | NetworkSlice2_Subnet | RingFence5_SecurityGroup |
| NetworkResource11_VM | NetworkSlicing2_VPC | NetworkSlice2_Subnet | RingFence5_SecurityGroup |
| NetworkResource12_VM | NetworkSlicing3_VPC | NetworkSlice3_Subnet | RingFence6_SecurityGroup |
| NetworkResource13_VM | NetworkSlicing3_VPC | NetworkSlice3_Subnet | RingFence6_SecurityGroup |
| NetworkResource14_VM | NetworkSlicing3_VPC | NetworkSlice3_Subnet | RingFence6_SecurityGroup |
| NetworkResource15_VM | NetworkSlicing3_VPC | NetworkSlice3_Subnet | RingFence6_SecurityGroup |
| NetworkResource16_VM | NetworkSlicing4_VPC | NetworkSlice4_Subnet | RingFence7_SecurityGroup |
| NetworkResource17_VM | NetworkSlicing4_VPC | NetworkSlice4_Subnet | RingFence7_SecurityGroup |
| NetworkResource18_VM | NetworkSlicing4_VPC | NetworkSlice4_Subnet | RingFence7_SecurityGroup |
| NetworkResource19_VM | NetworkSlicing4_VPC | NetworkSlice4_Subnet | RingFence8_SecurityGroup |
| NetworkResource20_VM | NetworkSlicing4_VPC | NetworkSlice4_Subnet | RingFence8_SecurityGroup |
| NetworkResource21_VM | NetworkSlicing4_VPC | NetworkSlice4_Subnet | RingFence8_SecurityGroup |
| NetworkResource22_VM | NetworkSlicing4_VPC | NetworkSlice4_Subnet | RingFence8_SecurityGroup |
| NetworkResource23_VM | NetworkSlicing5_VPC | NetworkSlice5_Subnet | RingFence9_SecurityGroup |
| NetworkResource24_VM | NetworkSlicing5_VPC | NetworkSlice5_Subnet | RingFence9_SecurityGroup |
| NetworkResource25_VM | NetworkSlicing5_VPC | NetworkSlice5_Subnet | RingFence10_SecurityGroup |
| NetworkResource26_VM | NetworkSlicing5_VPC | NetworkSlice5_Subnet | RingFence10_SecurityGroup |
| NetworkResource27_VM | NetworkSlicing5_VPC | NetworkSlice5_Subnet | RingFence10_SecurityGroup |
| NetworkResource28_VM | NetworkSlicing5_VPC | NetworkSlice5_Subnet | RingFence11_SecurityGroup |
| NetworkResource29_VM | NetworkSlicing5_VPC | NetworkSlice5_Subnet | RingFence11_SecurityGroup |

7. Go back to the 'Security Groups' dashboard. Click on 'RingFence1_SecurityGroup' and go to 'Inbound Rules' > 'Edit inbound rules'. Configure three inbound rules as type 'HTTP', 'HTTPS' and 'custom TCP' (1024-65535) for each ip of 'NetworkResource4_VM'. For each respective security group till 'RingFence11_SecurityGroup', ensure the following configuration is met as per the table:

Table 2: Ring Fence Security Group Configurations

| Ring Fence Security Group | Configuration Details |
|---|---|
| RingFence2_SecurityGroup | Configured with the IPs of NetworkResource1_VM, NetworkResource6_VM, and NetworkResource8_VM. |
| RingFence3_SecurityGroup | Configured with the IPs of NetworkResource3_VM and NetworkResource22_VM. |
| RingFence4_SecurityGroup | Configured with the IPs of NetworkResource1_VM, NetworkResource10_VM, and NetworkResource4_VM. |
| RingFence5_SecurityGroup | Configured with the IPs of NetworkResource8_VM, NetworkResource18_VM, and NetworkResource24_VM. |
| RingFence6_SecurityGroup | Configured with the source as RingFence6_SecurityGroup. |
| RingFence7_SecurityGroup | Configured with the IPs of NetworkResource10_VM, NetworkResource19_VM, NetworkResource22_VM, and NetworkResource23_VM. |
| RingFence8_SecurityGroup | Configured with the IPs of NetworkResource5_VM, NetworkResource17_VM, and NetworkResource18_VM. |
| RingFence9_SecurityGroup | Configured with the IPs of NetworkResource10_VM, NetworkResource16_VM, and NetworkResource26_VM. |
| RingFence10_SecurityGroup | Configured with the IP of NetworkResource24_VM. |
| RingFence11_SecurityGroup | Configured with the source as RingFence11_SecurityGroup. |

8. The security groups for the instances for their respective subnets in their VPC's are now configured. Attached below are images specific to the configuration in this manual.



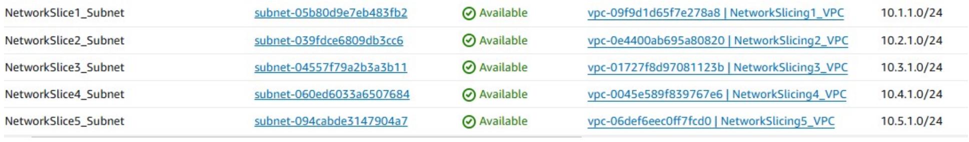
Your VPCs (6) Info

less

Search

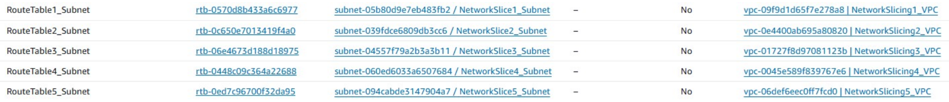
| Name | VPC ID | State | Block Public Access | IPv4 CIDR | IPv6 CIDR |
|---|---|---|---|---|---|
| – | vpc-0378e797170b373fa | Available | Off | 172.31.0.0/16 | – |
| NetworkSlicing1_VPC | vpc-09f9d1d65f7e278a8 | Available | Off | 10.1.0.0/16 | – |
| NetworkSlicing2_VPC | vpc-0e4400ab695a80820 | Available | Off | 10.2.0.0/16 | – |
| NetworkSlicing3_VPC | vpc-01727f8d97081123b | Available | Off | 10.3.0.0/16 | – |
| NetworkSlicing4_VPC | vpc-0045e589f839767e6 | Available | Off | 10.4.0.0/16 | – |
| NetworkSlicing5_VPC | vpc-06def6eec0ff7fcd0 | Available | Off | 10.5.0.0/16 | – |

Figure 5: VPC Dashboard for the created VPC's



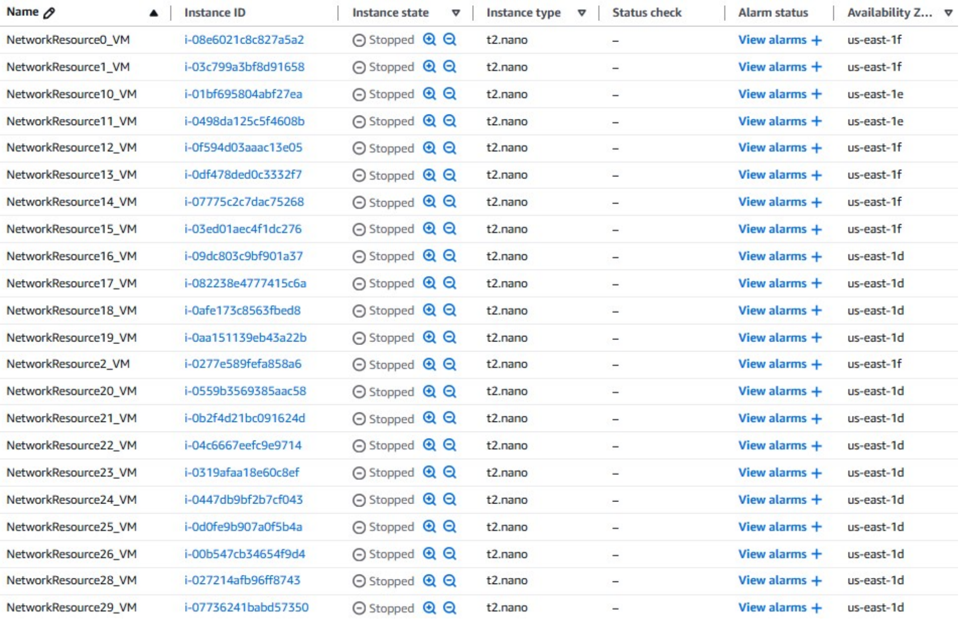
| | | | | |
|---|---|---|---|---|
| NetworkSlice1_Subnet | subnet-05b80d9e7eb483fb2 | Available | vpc-09f9d1d65f7e278a8 \| NetworkSlicing1_VPC | 10.1.1.0/24 |
| NetworkSlice2_Subnet | subnet-039fdce6809db3cc6 | Available | vpc-0e4400ab695a80820 \| NetworkSlicing2_VPC | 10.2.1.0/24 |
| NetworkSlice3_Subnet | subnet-04557f79a2b3a3b11 | Available | vpc-01727f8d97081123b \| NetworkSlicing3_VPC | 10.3.1.0/24 |
| NetworkSlice4_Subnet | subnet-060ed6033a6507684 | Available | vpc-0045e589f839767e6 \| NetworkSlicing4_VPC | 10.4.1.0/24 |
| NetworkSlice5_Subnet | subnet-094cabde3147904a7 | Available | vpc-06def6eec0ff7fcd0 \| NetworkSlicing5_VPC | 10.5.1.0/24 |

Figure 6: Subnet Dashboard for the created Subnets



| | | | | | |
|---|---|---|---|---|---|
| RouteTable1_Subnet | rtb-0570d8b433a6c6977 | subnet-05b80d9e7eb483fb2 / NetworkSlice1_Subnet | – | No | vpc-09f9d1d65f7e278a8 \| NetworkSlicing1_VPC |
| RouteTable2_Subnet | rtb-0c650e7013419f4a0 | subnet-039fdce6809db3cc6 / NetworkSlice2_Subnet | – | No | vpc-0e4400ab695a80820 \| NetworkSlicing2_VPC |
| RouteTable3_Subnet | rtb-06e4673d188d18975 | subnet-04557f79a2b3a3b11 / NetworkSlice3_Subnet | – | No | vpc-01727f8d97081123b \| NetworkSlicing3_VPC |
| RouteTable4_Subnet | rtb-0448c09c364a22688 | subnet-060ed6033a6507684 / NetworkSlice4_Subnet | – | No | vpc-0045e589f839767e6 \| NetworkSlicing4_VPC |
| RouteTable5_Subnet | rtb-0ed7c96700f32da95 | subnet-094cabde3147904a7 / NetworkSlice5_Subnet | – | No | vpc-06def6eec0ff7fcd0 \| NetworkSlicing5_VPC |

Figure 7: Route Table Dashboard for the created Route Tables



| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Z... |
|---|---|---|---|---|---|---|
| NetworkResource0_VM | i-08e6021c8c827a5a2 | Stopped | t2.nano | – | View alarms + | us-east-1f |
| NetworkResource1_VM | i-03c799a3bf8d91658 | Stopped | t2.nano | – | View alarms + | us-east-1f |
| NetworkResource10_VM | i-01bf695804abf27ea | Stopped | t2.nano | – | View alarms + | us-east-1e |
| NetworkResource11_VM | i-0498da125c5f4608b | Stopped | t2.nano | – | View alarms + | us-east-1e |
| NetworkResource12_VM | i-0f594d03aaac13e05 | Stopped | t2.nano | – | View alarms + | us-east-1f |
| NetworkResource13_VM | i-0df478ded0c3332f7 | Stopped | t2.nano | – | View alarms + | us-east-1f |
| NetworkResource14_VM | i-07775c2c7dac75268 | Stopped | t2.nano | – | View alarms + | us-east-1f |
| NetworkResource15_VM | i-03ed01aec4f1dc276 | Stopped | t2.nano | – | View alarms + | us-east-1f |
| NetworkResource16_VM | i-09dc803c9bf901a37 | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource17_VM | i-082238e4777415c6a | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource18_VM | i-0afe173c8563fbed8 | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource19_VM | i-0aa151139eb43a22b | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource2_VM | i-0277e589fefa858a6 | Stopped | t2.nano | – | View alarms + | us-east-1f |
| NetworkResource20_VM | i-0559b3569385aac58 | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource21_VM | i-0b2f4d21bc091624d | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource22_VM | i-04c6667eefc9e9714 | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource23_VM | i-0319afaa18e60c8ef | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource24_VM | i-0447db9bf2b7cf043 | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource25_VM | i-0d0fe9b907a0f5b4a | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource26_VM | i-00b547cb34654f9d4 | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource28_VM | i-027214afb96ff8743 | Stopped | t2.nano | – | View alarms + | us-east-1d |
| NetworkResource29_VM | i-07736241babd57350 | Stopped | t2.nano | – | View alarms + | us-east-1d |

Figure 8: EC2 Instance Dashboard for the created Instances

9. Go to the 'VPC Dashboard' and click on 'create endpoint'. For this configurational purpose, the first VPC endpoint shall be titled 'EP_NetworkSlicing1_VPC. Select 'AWS services' and 'com.amazonaws.us-east-1.s3' as the service with Type 'Gateway'. Select 'NetworkSlicing1_VPC' as the VPC and select 'RouteTable1_Subnet' as the Subnet.

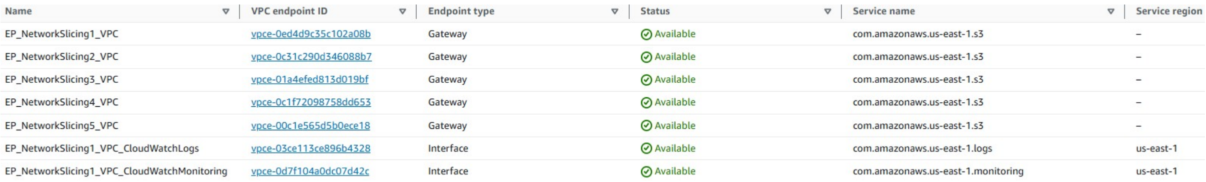
Click on ’Custom’ in Policy and copy the JSON policy from the text document labelled ’JSON for VPC Endpoint EP_NetworkSlicing1_VPC to EP_NetworkSlicing5_VPC’ to this ’Custom’ Policy. Create the endpoint. Repeat step 9 for ’EP_NetworkSlicing2_VPC’ to ’EP_NetworkSlicing5_VPC’. Then create another VPC endpoint with the title ’EP_NetworkSlicing1_VPC_CloudWatchLogs’ for this configurational purpose. Select ’AWS services’ and ’com.amazonaws.us-east-1.s3’ as the service with Type ’Interface’. Select ’NetworkSlicing1_VPC’ as the VPC and select the subnet that is available. Ensure ’Enable DNS name’ is checked and select the available subnet. Select the security groups ’RingFence1_SecurityGroup’ and ’RingFence2_SecurityGroup’. Click on ’Custom’ in Policy and copy the JSON policy from the text document labelled ’JSON for VPC Endpoint CloudWatchLogs and CloudWatchMonitoring’. Create the VPC endpoint ’EP_NetworkSlicing1_VPC_CloudWatchLogs’. Create another VPC endpoint with the title ’EP_NetworkSlicing1_VPC_CloudWatchMonitoring’ with the same procedure as the VPC endpoint ’EP_NetworkSlicing1_VPC_CloudWatchLogs’.

10. Go to the ’VPC Peering’ dashboard. Click on ’Create peering connection’. For this configurational purpose, the first VPC Peering endpoint shall be titled ’NetworkSlice1_NetworkSlice2_VPCPeering’. Select ’NetworkSlicing1_VPC’ and ’NetworkSlicing2_VPC’ as the Requester and Acceptor. Click on ’Create peering connection’. Go to Actions > ’Accept request’ to accept the VPC peering connection.
Repeat Step 10 for the following VPC Peering connections titled ’NetworkSlice2_NetworkSlice5_VPCPeering’, ’NetworkSlice2_NetworkSlice4_VPCPeering’ and ’NetworkSlice4_NetworkSlice5_VPCPeering’ for this configurational example. Note that the names of the ’NetworkSlice_VPCPeering’ VPC peering endpoints mentioned above shall correspond to that Requester and Acceptor.
11. Go to the ’Internet Gateway’ dashboard. Click on ’Create internet gateway’. For this configurational purpose, the first internet gateway shall be titled as ’IG_NetworkSlicing_1’. Go to ’Actions’ >’Attach to VPC’ and select ’NetworkSlicing1_VPC’. Click on ’Attach internet gateway’. Repeat Step 11 for internet gateways ’IG_NetworkSlicing_2’ to ’IG_NetworkSlicing_5’.
12. Go to the ’Elastic IP’s’ dashboard. Click on ’Allocate Elastic IP address’ and then click on ’Allocate’. Repeat this step four times for five elastic IP’s.
13. Go to the ’NAT Gateways’ dashboard. Click on ’Create NAT gateway’. For this configurational purpose, the first ’NAT gateway’ shall be titled ’NG_NetworkSlicing_1’. Select ’NetworkSlice1_Subnet’, allocate an elastic ip and create the NAT gateway. Repeat step 13 for four more NAT gateways, namely ’NG_NetworkSlicing_2’ to ’NG_NetworkSlicing_5’ for ’NetworkSlice2_Subnet’ to ’NetworkSlice5_Subnet with an elastic ip each.
14. Go to the ’Route Tables’ dashboard and click on ’RouteTable1_Subnet’. Go to ’Routes’ and click on ’Edit routes’. Configure three routes for ’NetworkSlice1_NetworkSlice2_VPCPeering’, ’NG_NetworkSlicing_1’ and ’EP_NetworkSlicing1_VPC’ as targets with 10.2.0.0/16, 0.0.0.0/0 and ’com.amazonaws.us-east-1.s3’ (Endpoint type - Gateway) set as ’Destination’ respectively. Similarly, set the ’Route Table’ configuration for the route tables below.

Table 3: Route Table Configurations for Subnets

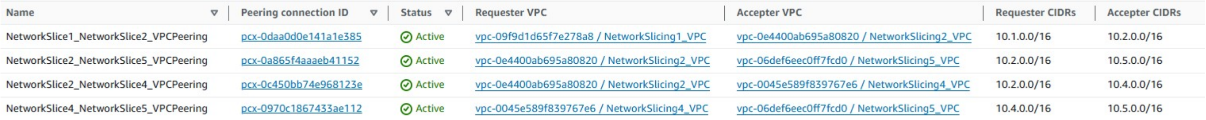
| Route Table | Target | Destination |
|---|---|---|
| RouteTable2_Subnet | NetworkSlice1_NetworkSlice2_VPCPeering | 10.1.0.0/16 |
| | NetworkSlice2_NetworkSlice4_VPCPeering | 10.4.0.0/16 |
| | NetworkSlice2_NetworkSlice5_VPCPeering | 10.5.0.0/16 |
| | NG_NetworkSlicing_2 | 0.0.0.0/0 |
| | com.amazonaws.us-east-1.s3 (Gateway) | com.amazonaws.us-east-1.s3 |
| RouteTable3_Subnet | NG_NetworkSlicing_3 | 0.0.0.0/0 |
| | com.amazonaws.us-east-1.s3 (Gateway) | com.amazonaws.us-east-1.s3 |
| RouteTable4_Subnet | NetworkSlice2_NetworkSlice4_VPCPeering | 10.2.0.0/16 |
| | NetworkSlice4_NetworkSlice5_VPCPeering | 10.5.0.0/16 |
| | NG_NetworkSlicing_4 | 0.0.0.0/0 |
| | EP_NetworkSlicing4_VPC | com.amazonaws.us-east-1.s3 |
| RouteTable5_Subnet | NetworkSlice2_NetworkSlice5_VPCPeering | 10.2.0.0/16 |
| | NetworkSlice4_NetworkSlice5_VPCPeering | 10.4.0.0/16 |
| | NG_NetworkSlicing_5 | 0.0.0.0/0 |
| | EP_NetworkSlicing4_VPC | com.amazonaws.us-east-1.s3 |

15. The inter-VM connections and connection to the internet cloud are now established for the instances for their respective subnets in their VPC's. Attached below are images specific to the configuration in this manual.



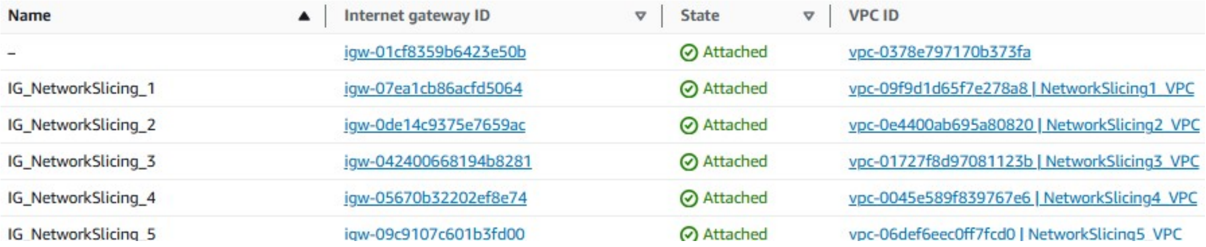
| Name | VPC endpoint ID | Endpoint type | Status | Service name | Service region |
|---|---|---|---|---|---|
| EP_NetworkSlicing1_VPC | vpce-0ed4d9c35c102a08b | Gateway | ✓ Available | com.amazonaws.us-east-1.s3 | – |
| EP_NetworkSlicing2_VPC | vpce-0c31c290d346088b7 | Gateway | ✓ Available | com.amazonaws.us-east-1.s3 | – |
| EP_NetworkSlicing3_VPC | vpce-01a4efed813d019bf | Gateway | ✓ Available | com.amazonaws.us-east-1.s3 | – |
| EP_NetworkSlicing4_VPC | vpce-0c1f72098758dd653 | Gateway | ✓ Available | com.amazonaws.us-east-1.s3 | – |
| EP_NetworkSlicing5_VPC | vpce-00c1e565d5b0ece18 | Gateway | ✓ Available | com.amazonaws.us-east-1.s3 | – |
| EP_NetworkSlicing1_VPC_CloudWatchLogs | vpce-03ce113ce896b4328 | Interface | ✓ Available | com.amazonaws.us-east-1.logs | us-east-1 |
| EP_NetworkSlicing1_VPC_CloudWatchMonitoring | vpce-0d7f104a0dc07d42c | Interface | ✓ Available | com.amazonaws.us-east-1.monitoring | us-east-1 |

Figure 9: VPC Endpoints Dashboard for the created VPC Endpoints



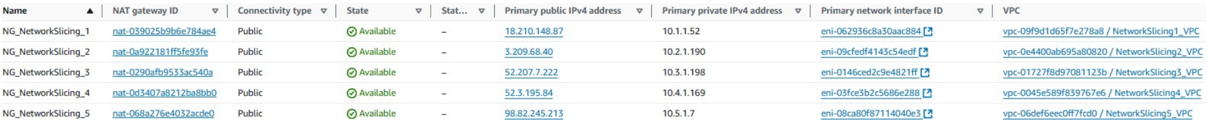
| Name | Peering connection ID | Status | Requester VPC | Accepter VPC | Requester CIDRs | Accepter CIDRs |
|---|---|---|---|---|---|---|
| NetworkSlice1_NetworkSlice2_VPCPeering | pcx-0daa0d0e141a1e385 | ✓ Active | vpc-09f9d1d65f7e278a8 / NetworkSlicing1_VPC | vpc-0e4400ab695a80820 / NetworkSlicing2_VPC | 10.1.0.0/16 | 10.2.0.0/16 |
| NetworkSlice2_NetworkSlice5_VPCPeering | pcx-0a865f4aaaeb41152 | ✓ Active | vpc-0e4400ab695a80820 / NetworkSlicing2_VPC | vpc-06def6eec0ff7fcd0 / NetworkSlicing5_VPC | 10.2.0.0/16 | 10.5.0.0/16 |
| NetworkSlice2_NetworkSlice4_VPCPeering | pcx-0c450bb74e968123e | ✓ Active | vpc-0e4400ab695a80820 / NetworkSlicing2_VPC | vpc-0045e589f839767e6 / NetworkSlicing4_VPC | 10.2.0.0/16 | 10.4.0.0/16 |
| NetworkSlice4_NetworkSlice5_VPCPeering | pcx-0970c1867433ae112 | ✓ Active | vpc-0045e589f839767e6 / NetworkSlicing4_VPC | vpc-06def6eec0ff7fcd0 / NetworkSlicing5_VPC | 10.4.0.0/16 | 10.5.0.0/16 |

Figure 10: VPC Peering Dashboard for the active VPC Peering Connections



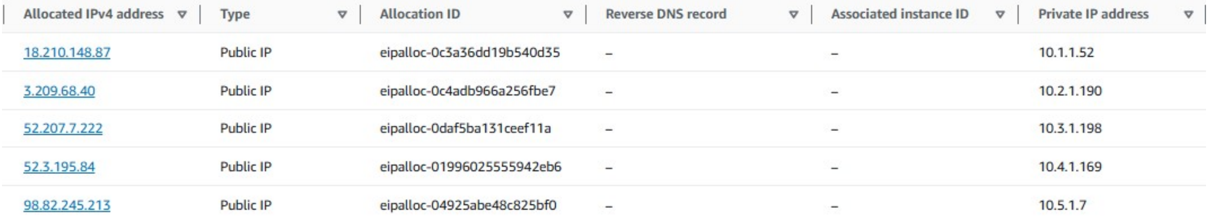
| Name | Internet gateway ID | State | VPC ID |
|---|---|---|---|
| – | igw-01cf8359b6423e50b | ✓ Attached | vpc-0378e797170b373fa |
| IG_NetworkSlicing_1 | igw-07ea1cb86acfd5064 | ✓ Attached | vpc-09f9d1d65f7e278a8 \| NetworkSlicing1_VPC |
| IG_NetworkSlicing_2 | igw-0de14c9375e7659ac | ✓ Attached | vpc-0e4400ab695a80820 \| NetworkSlicing2_VPC |
| IG_NetworkSlicing_3 | igw-042400668194b8281 | ✓ Attached | vpc-01727f8d97081123b \| NetworkSlicing3_VPC |
| IG_NetworkSlicing_4 | igw-05670b32202ef8e74 | ✓ Attached | vpc-0045e589f839767e6 \| NetworkSlicing4_VPC |
| IG_NetworkSlicing_5 | igw-09c9107c601b3fd00 | ✓ Attached | vpc-06def6eec0ff7fcd0 \| NetworkSlicing5_VPC |

Figure 11: Internet Gateway Dashboard for the created Internet Gateways



| Name | NAT gateway ID | Connectivity type | State | Stat... | Primary public IPv4 address | Primary private IPv4 address | Primary network interface ID | VPC |
|---|---|---|---|---|---|---|---|---|
| NG_NetworkSlicing_1 | nat-039025b9b6e784ae4 | Public | ✓ Available | – | 18.210.148.87 | 10.1.1.52 | eni-062936c8a30aac884 | vpc-09f9d1d65f7e278a8 / NetworkSlicing1_VPC |
| NG_NetworkSlicing_2 | nat-0a922181ff5fe93fe | Public | ✓ Available | – | 3.209.68.40 | 10.2.1.190 | eni-09cfedf4143c54edf | vpc-0e4400ab695a80820 / NetworkSlicing2_VPC |
| NG_NetworkSlicing_3 | nat-0290afb9533ac540a | Public | ✓ Available | – | 52.207.7.222 | 10.3.1.198 | eni-0146ced2c9e4821ff | vpc-01727f8d97081123b / NetworkSlicing3_VPC |
| NG_NetworkSlicing_4 | nat-0d3407a8212ba8bb0 | Public | ✓ Available | – | 52.3.195.84 | 10.4.1.169 | eni-03fce3b2c5686e288 | vpc-0045e589f839767e6 / NetworkSlicing4_VPC |
| NG_NetworkSlicing_5 | nat-068a276e4032acde0 | Public | ✓ Available | – | 98.82.245.213 | 10.5.1.7 | eni-08ca80f87114040e3 | vpc-06def6eec0ff7fcd0 / NetworkSlicing5_VPC |

Figure 12: Nat Gateway Dashboard for the created NAT Gateways



| Allocated IPv4 address | Type | Allocation ID | Reverse DNS record | Associated instance ID | Private IP address |
|---|---|---|---|---|---|
| 18.210.148.87 | Public IP | eipalloc-0c3a36dd19b540d35 | – | – | 10.1.1.52 |
| 3.209.68.40 | Public IP | eipalloc-0c4adb966a256fbe7 | – | – | 10.2.1.190 |
| 52.207.7.222 | Public IP | eipalloc-0daf5ba131ceef11a | – | – | 10.3.1.198 |
| 52.3.195.84 | Public IP | eipalloc-01996025555942eb6 | – | – | 10.4.1.169 |
| 98.82.245.213 | Public IP | eipalloc-04925abe48c825bf0 | – | – | 10.5.1.7 |

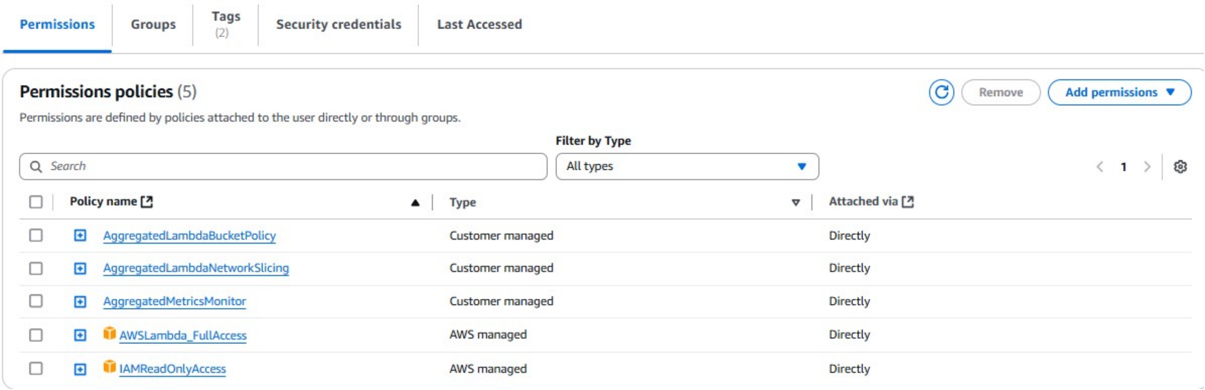
Figure 13: Elastic IP Dashboard for the created Elastic IP's

16. Go to the IAM Dashboard and click on 'Users'. Click on 'Add users'. For this configurational purpose, the IAM user shall be titled 'Aggregated-Lambda-Cli-User'. Click on Next and then click on 'Attach policies directly'. Attach the permission policies 'AWSLambda_FullAccess' and 'IAMReadOnlyAccess'. Click on Next. Click on 'Add new tag' and set 'Environment': 'Development' for the newly created tag. Create the user 'Aggregated-Lambda-Cli-User' and click on 'Aggregated-Lambda-Cli-User'. Go to 'Security Credentials' and click on 'Create access key'. Click on 'Command Line Interface(CLI)', check the 'Confirmation' and click on 'Next'.
Provide a suitable description to 'Aggregated-Lambda-Cli-User' and click on 'Next'. Download the .csv file containing the IAM Access Key and Secret Access Key. Note down the Secret Access Key as this secret access key shall be used to configure the AWS CLI from the terminal in Eclipse IDE. Go to 'Tags' and click on 'Manage Tags'. Ensure your 'Secret Access Key' with the suitable description is added there as a tag along with 'Environment': 'Development'.
17. Go to Policies in the IAM Dashboard and click on 'Create policy'. Click on 'JSON' and open the text document titled 'JSON for AggregatedLambdaBucketPolicy'. Copy the JSON policy from 'JSON for AggregatedLambdaBucketPolicy' to 'Policy Editor' under 'JSON' and click on 'Next'. For this configurational purpose, the policy shall be titled 'AggregatedLambdaBucketPolicy'. Provide a suitable description to 'AggregatedLambdaBucketPolicy' and click on 'Create policy'. Repeat Step 17 with the policies titled 'AggregatedLambdaNetworkSlicing' and 'AggregatedMetricsMonitor' with the JSON policies from 'JSON for AggregatedLambdaNetworkSlicing' and 'JSON for AggregatedMetricsMonitor' respectively.
18. Go to Roles in the IAM dashboard and click on 'Create role'. For the 'Trusted entity type', choose 'AWS service' and select 'Lambda' for 'Service or use case'. Click on 'Next'. Attach 'AggregatedLambdaBucketPolicy', 'AggregatedLambdaNetworkSlicing' and 'AggregatedMetricsMonitor' as 'Permission policies' and click on 'Next'. For this configurational purpose, the IAM role shall be titled 'AggregatedLambdaExecutionRole'. Create 'AggregatedLambdaExecutionRole' and navigate to the 'Trust relationships' tab. Verify that the JSON policy matches the policy as per the text document 'JSON Trust Relationships for AggregatedLambdaNetworkSlicing'. Go back to Policies in the IAM Dashboard and attach 'Aggregated-Lambda-Cli-User' with 'AggregatedLambdaExecutionRole' for 'AggregatedLambdaBucketPolicy', 'AggregatedLambdaNetworkSlicing' and 'AggregatedMetricsMonitor'. Go back to Users in the IAM Dashboard and click on 'Aggregated-Lambda-Cli-User'. Go to the 'Permissions' tab and click on 'Add permissions' >'Attach policies directly'. Attach the policies 'AggregatedLambdaBucketPolicy', 'AggregatedLambdaNetworkSlicing' and 'AggregatedMetricsMonitor' to 'Aggregated-Lambda-Cli-User'. Click on 'Add permissions'.
19. Go to the Lambda Dashboard and click on 'Create function'. For this configurational purpose, the Lambda function shall be titled 'AggregatedLambdaNetworkSlicing'.

Select 'Java 17' as the runtime and click on 'Additional Configurations'. Check 'Enable VPC' and select 'NetworkSlicing1_VPC' with 'NetworkSlice1_Subnet' as the VPC and Subnet. Select 'RingFence1_SecurityGroup' and 'RingFence2_SecurityGroup' as the security groups. Click on 'Create function'. Go to the 'Configuration' tab and click on 'General Configuration' in 'AggregatedLambdaNetworkSlicing'. Click on 'Edit' and adjust 'Memory' to 2048 MB, 'Timeout' to 7 minutes and 'Existing role' to 'AggregatedLambdaExecutionRole'. Click on 'Save'.

20. Go to the Amazon S3 Dashboard and click on 'create bucket'. For this configurational purpose, the S3 bucket shall be titled 'aggregatedlambdabucketoutput'. Click on 'Create bucket'. Upload the 'preprocessed_fastStorage.csv' and 'preprocessed_rnd.csv' from the 'Research Project/GWA-T-12BitbrainsPreprocessing' folder in your machine.

21. The IAM user, role and policies is now configured to use the newly created Lambda function to upload each CloudSim Project as a packaged fat JAR/Zip file using AWS CLI from the terminal in Eclipse IDE. The simulation outputs for each CloudSim Project shall be stored in the S3 Bucket respectively. Attached below are images specific to the configuration in this manual.



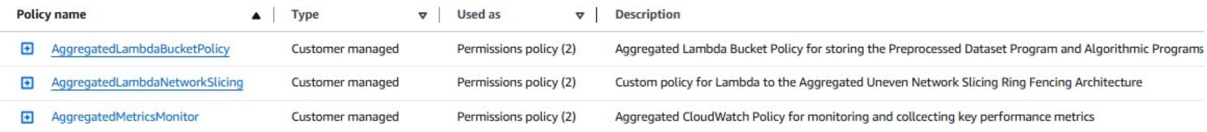
Permissions policies (5)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

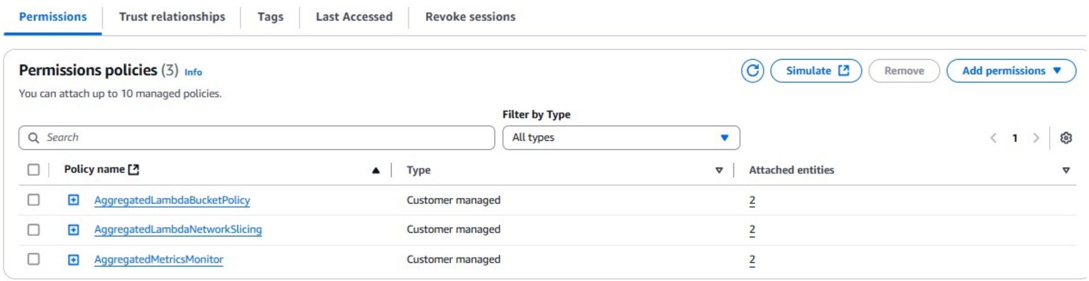
| Policy name | Type | Attached via |
|---|---|---|
| AggregatedLambdaBucketPolicy | Customer managed | Directly |
| AggregatedLambdaNetworkSlicing | Customer managed | Directly |
| AggregatedMetricsMonitor | Customer managed | Directly |
| AWSLambda_FullAccess | AWS managed | Directly |
| IAMReadOnlyAccess | AWS managed | Directly |

Figure 14: IAM User Dashboard for the newly created user to access IAM-related details, Lambda, S3 Bucket and CloudWatch.



| Policy name | Type | Used as | Description |
|---|---|---|---|
| AggregatedLambdaBucketPolicy | Customer managed | Permissions policy (2) | Aggregated Lambda Bucket Policy for storing the Preprocessed Dataset Program and Algorithmic Programs |
| AggregatedLambdaNetworkSlicing | Customer managed | Permissions policy (2) | Custom policy for Lambda to the Aggregated Uneven Network Slicing Ring Fencing Architecture |
| AggregatedMetricsMonitor | Customer managed | Permissions policy (2) | Aggregated CloudWatch Policy for monitoring and collcecting key performance metrics |

Figure 15: IAM Policies Dashboard for the newly created policies facilitating access to Lambda, S3 Bucket and CloudWatch by the user.



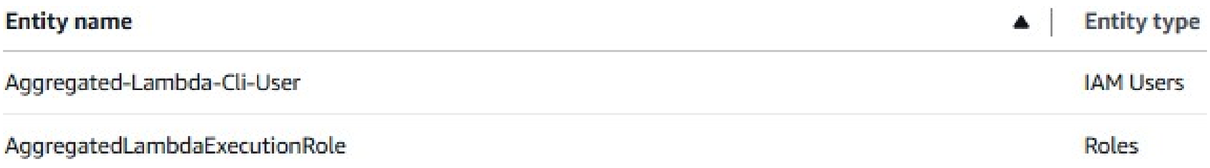
Permissions policies (3) Info

You can attach up to 10 managed policies.

Filter by Type: All types

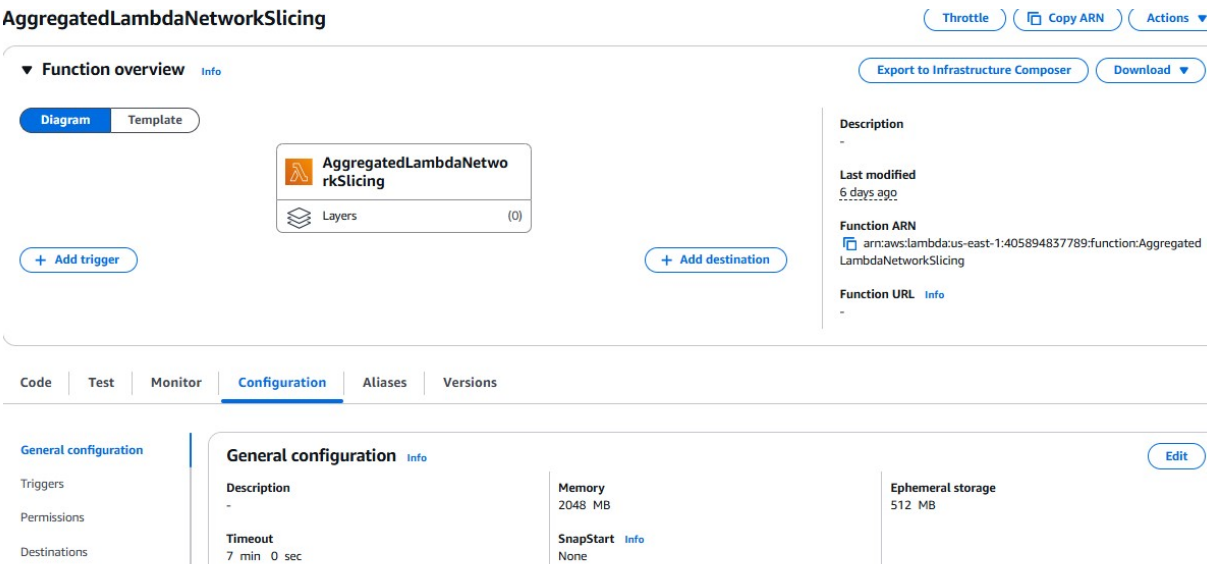
| Policy name | Type | Attached entities |
|---|---|---|
| AggregatedLambdaBucketPolicy | Customer managed | 2 |
| AggregatedLambdaNetworkSlicing | Customer managed | 2 |
| AggregatedMetricsMonitor | Customer managed | 2 |

Figure 16: IAM Role Dashboard for the newly created IAM role on the IAM user for S3 Bucket and CloudWatch.



| Entity name | Entity type |
|---|---|
| Aggregated-Lambda-Cli-User | IAM Users |
| AggregatedLambdaExecutionRole | Roles |

Figure 17: IAM Policies Dashboard for the newly created IAM policies for S3 Bucket and CloudWatch used by the IAM user.



Description
-

Last modified
6 days ago

Function ARN
arn:aws:lambda:us-east-1:405894837789:function:AggregatedLambdaNetworkSlicing

Function URL Info
-

General configuration Info

Description
-

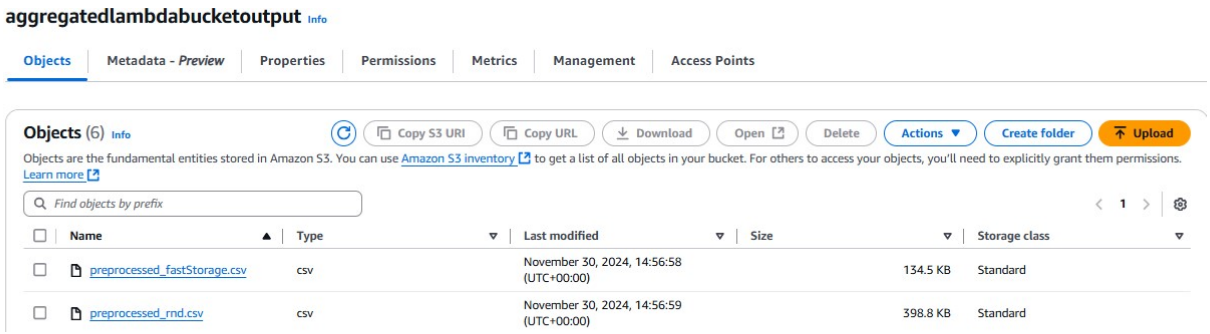
Memory
2048 MB

Ephemeral storage
512 MB

Timeout
7 min 0 sec

SnapStart Info
None

Figure 18: Lambda Dashboard for the newly Lambda function to be used by the IAM user.



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

| □ | Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|---|
| □ | preprocessed_fastStorage.csv | csv | November 30, 2024, 14:56:58 (UTC+00:00) | 134.5 KB | Standard |
| □ | preprocessed_rnd.csv | csv | November 30, 2024, 14:56:59 (UTC+00:00) | 398.8 KB | Standard |

Figure 19: S3 Bucket Dashboard for the logs to store the simulation outputs for aggregated algorithmic program.

22. Go to Eclipse IDE and right-click on Package explorer and click on >'Java Project'. For the 'iterative heuristic energy-aware non-convex' aggregated algorithmic program in this configurational purpose, the 'CloudSim' Project shall be titled 'lambdafinaliterativeresearchprojectfinal'. Click on 'Next' and then 'Finish'. Move into the directory 'lambdafinaliterativeresearchprojectfinal' in Eclipse IDE and enter the following command below:

```
aws configure
```

Provide the Access Key ID, Secret Access Key from the .csv file downloaded earlier. Since this configurational example's architectural environment is in us-east-1, select 'us-east-1' as the default region. Right-click on click on 'Configure' >'Convert to Maven Project'. Click on 'Finish'. Enter the following command below in the terminal on Eclipse IDE:

```
mvn install:install-file -Dfile=C:/Users/<username>/Cloudsim_Research
Project/cloudsim-3.0.3/jars/cloudsim-3.0.3.jar -DgroupId=org.cloudbus.
cloudsim -DartifactId=cloudsim -Dversion=3.0.3 -Dpackaging=jar -DpomFile=
C:/Users/<username>/Cloudsim_ResearchProject/cloudsim-3.0.3/jars/
cloudsim-3.0.3.pom
```

Click on the created pom.xml created in 'lambdafinaliterativeresearchprojectfinal'. Copy and paste the contents of the text document 'pom.xml for Aggregated Algorithmic Programs on AWS' into the newly created pom.xml in 'lambdafinaliterativeresearchprojectfinal'. Enter the following command below in the terminal on Eclipse IDE.

```
mvn clean install -U
```

Navigate to the src folder in 'lambdafinaliterativeresearchprojectfinal' and delete module-info.java. Right-click on src/main/java in 'lambdafinaliterativeresearchprojectfinal' and click on 'Build Path' >'Remove from Build Path'. Delete the src folder. Right-click on the project folder 'lambdafinaliterativeresearchprojectfinal' and click on 'New' >'Source Folder'. Name the source folder as 'src/main/java'. Right-click on the project folder 'lambdafinaliterativeresearchprojectfinal' and click on 'New' >'Class'. Name the class 'FinalAggregatedIterativeHeuristicNonConvexEnergyAwareLambda'. Open the text document labelled 'FinalAggregatedIterativeHeuristicNonConvexEnergyAwareLambda.java' and copy the packages from the text document 'FinalAggregatedIterativeHeuristicNonConvexEnergyAwareLambda.java' into the algorithmic program on Eclipse. Then copy the rest of the program from the text document 'FinalAggregatedIterativeHeuristicNonConvexEnergyAwareLambda.java' onto the algorithmic program on Eclipse. Save the program. Open the text document labelled 'Additional Dependencies for Aggregated Algorithmic Programs on AWS' on your machine and copy the extra dependencies to your pom.xml created in 'lambdafinaliterativeresearchprojectfinal'. Run the following command below on the terminal in Eclipse IDE:

```
mvn clean package
```

23. Right-click on the project folder 'lambdafinaliterativeresearchprojectfinal' >'Update Project'. Check 'Force Update of Snapshots/Releases' and click on OK. Enter the following commands on the terminal in the project folder lambdafinaliterativeresearchprojectfinal' in Eclipse IDE:

```
aws lambda update-function-code –function-name AggregatedLambdaNetwork
Slicing –zip-file fileb://C:/Users/<username>/Cloudsim_ResearchProject
<projectfolder>/target/FinalAggregatedIterativeHeuristicNon
ConvexEnergyAwareLambda-1.0-SNAPSHOT.jar
```

```
aws lambda update-function-configuration --function-name AggregatedLambda
NetworkSlicing --handler lambdafinaliterativeresearchprojectfinal.Final
FinalAggregatedIterativeHeuristicNonConvexEnergyAwareLambda::handleRequest

aws lambda get-function --function-name AggregatedLambdaNetworkSlicing
```

24. Go to the 'AggregatedLambdaNetworkSlicing' Lambda function in your Lambda Dashboard and click on the configuration tab 'Test'. For this configurational purpose, the test event shall be titled 'AggregatedIterLambdaTest'. Open the text document 'JSON for AggregatedLambdaNetworkSlicing AggregatedIterLambdaTest' and copy the JSON to the Event JSON in 'AggregatedLambdaNetworkSlicing' Lambda on the Lambda dashboard. The Lambda function 'AggregatedLambdaNetworkSlicing' is now ready to run. Run the test event 'AggregatedIterLambdaTest' and click on 'Logs' after the event is run. In the CLoudWatch Dashboard, click on 'All metrics' and then click on the newly created CloudWatch custom namespace 'AggregatedMetricsMonitor' >'Metrics with no dimensions'.
Click on 'Graphed Metrics' by selecting the 'performance metrics' specific to the graphical display and adjust the period if the 'AggregatedLambdaNetworkSlicing' Lambda was created far back. Repeat steps 22 and 23 exactly as per the configuration below for the other aggregated algorithmic programs which are 'lambdafinalheuristicresearchprojectfinal' and 'lambdafinalpriorityresearchprojectfinal' as per the table in this configurational example below. Replace the project folder name and class name when the AWS CLI commands are run as per the terminal on Eclipse IDE. The aws configure command does not need to be run again.

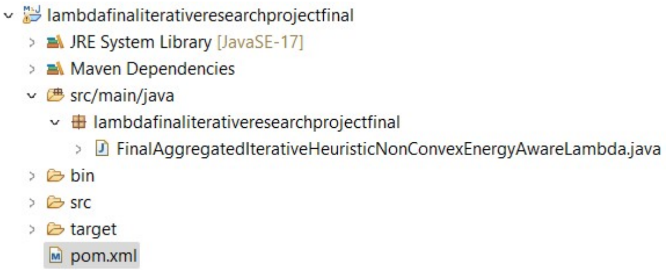
Table 4: Project Folder and Class Details

| 'Project Folder' | 'Class' |
|---|---|
| lambdafinalheuristicresearchprojectfinal | AggregatedHeuristicAugmentLambda.java |
| lambdafinalpriorityresearchprojectfinal | FinalAggregatedPriorityLambda.java |

Table 5: 'JSON for AggregatedHeuristicAugmentLambda Lambda' and Lambda Test Event Details

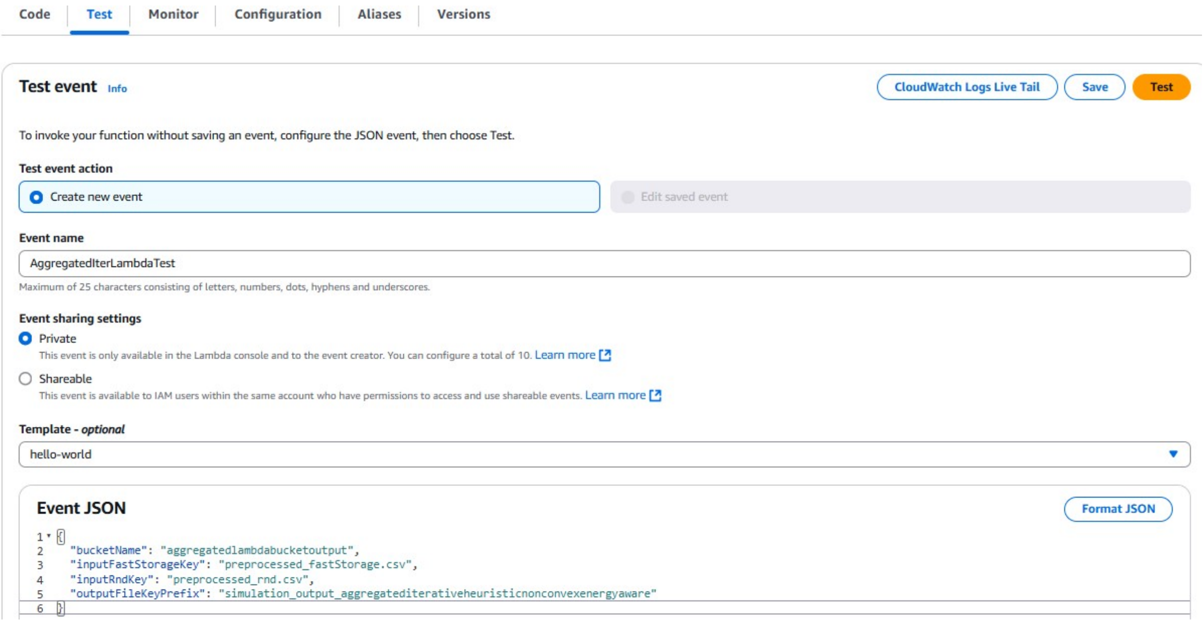
| JSON for AggregatedHeuristicAugmentLambda Lambda | Lambda Test Event Details |
|---|---|
| AggregatedLambdaNetworkSlicing AggregatedHeurLambdaTest | AggregatedHeurLambdaTest.java |
| AggregatedLambdaNetworkSlicing AggregatedPriLambdaTest | AggregatedPriLambdaTest.java |

Attached below are images specific to the configuration in this manual for the test event 'AggregatedIterLambdaTest' run on 'AggregatedLambdaNetworkSlicing' Lambda.



```
lambdafinaliterativeresearchprojectfinal
  JRE System Library [JavaSE-17]
  Maven Dependencies
  src/main/java
    lambdafinaliterativeresearchprojectfinal
      FinalAggregatedIterativeHeuristicNonConvexEnergyAwareLambda.java
  bin
  src
  target
  pom.xml
```

Figure 20: Project Explorer of the Iterative Heuristic Energy-Aware Non-Convex Algorithmic program on Eclipse IDE.



**Test event** Info

To invoke your function without saving an event, configure the JSON event, then choose Test.

**Test event action**

● Create new event ○ Edit saved event

**Event name**

AggregatedIterLambdaTest

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

**Event sharing settings**

● Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

○ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

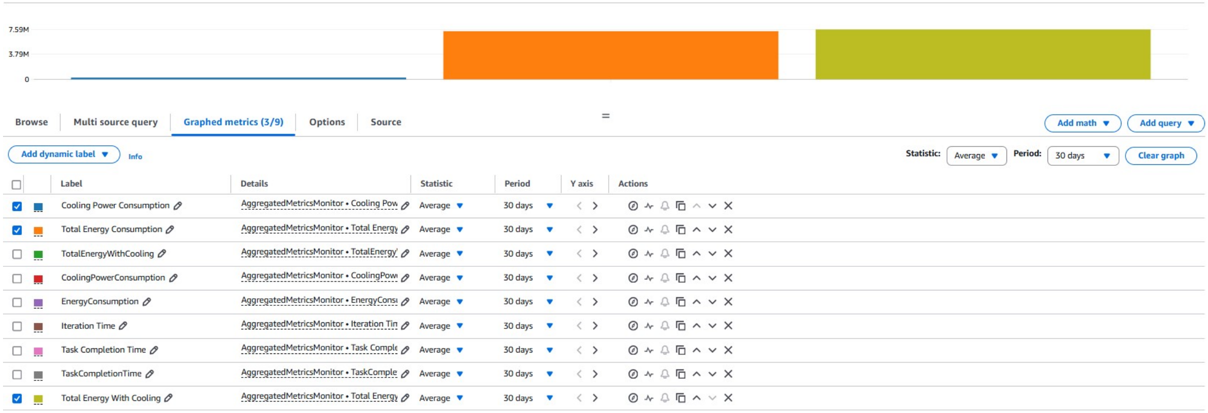
**Template - *optional***

hello-world

**Event JSON**

```
{
    "bucketName": "aggregatedlambdabucketoutput",
    "inputFastStorageKey": "preprocessed_fastStorage.csv",
    "inputRndKey": "preprocessed_rnd.csv",
    "outputFileKeyPrefix": "simulation_output_aggregatediterativeheuristicnonconvexenergyaware"
}
```

Figure 21: Lambda Test Event for the Iterative Heuristic Energy-Aware Non-Convex Algorithmic program on the Lambda function.



Statistic: Average Period: 30 days

| □ | | Label | Details | Statistic | Period | Y axis | Actions |
|---|---|---|---|---|---|---|---|
| ☑ | 🟦 | Cooling Power Consumption | AggregatedMetricsMonitor • Cooling Pow | Average | 30 days | | |
| ☑ | 🟧 | Total Energy Consumption | AggregatedMetricsMonitor • Total Energy | Average | 30 days | | |
| □ | 🟩 | TotalEnergyWithCooling | AggregatedMetricsMonitor • TotalEnergy' | Average | 30 days | | |
| □ | 🟥 | CoolingPowerConsumption | AggregatedMetricsMonitor • CoolingPow | Average | 30 days | | |
| □ | 🟪 | EnergyConsumption | AggregatedMetricsMonitor • EnergyConsı | Average | 30 days | | |
| □ | 🟫 | Iteration Time | AggregatedMetricsMonitor • Iteration Tin | Average | 30 days | | |
| □ | ■ | Task Completion Time | AggregatedMetricsMonitor • Task Comple | Average | 30 days | | |
| □ | ■ | TaskCompletionTime | AggregatedMetricsMonitor • TaskComple | Average | 30 days | | |
| ☑ | 🟨 | Total Energy With Cooling | AggregatedMetricsMonitor • Total Energy | Average | 30 days | | |

Figure 22: CloudWatch Monitoring for the Lambda function.

# 4 Implementation Notes

## 4.1 CloudSim Implementation and AWS Environmental Setup

### 4.1.1 CloudSim Environmental Setup

**Creation of CloudSim Environment**: The CloudSim environment was initialised for the preprocessed dataset program defining the number of users.
**Creation of Datacenters, Virtual Machine Allocation and Cloudlet Submission**: Next, created datacenters that consisted of hosts configured with CPU cores, RAM, network bandwidth and storage were incorporated as network slices and hosts as ring fences. Created VM's configured for MIPS(million instructions per second) as L3 computational network resource in ring fences were submitted to the Datacenter Broker.
Created workload tasks from preprocessed_fastStorage.csv and preprocessed_rnd.csv were simulated as workloads into VM's.
**Inter-VM Communication and Total Energy Consumption**: VM-to-VM communication links between the L3 computational network resources were incorporated using the NetworkTopology functionality. The cooling constraint was indirectly simulated through a shared constraint on bandwidth or energy and the total energy consumption with all logical components was calculated.
**Simulation Execution and Printed Results**: With the simulation finished here, the total energy consumption including the performance metrics were printed to the output log file for analysis.

### 4.1.2 AWS Environmental Setup

An architectural diagram representing the implementation of the aggregated uneven network slicing ring fencing architecture with associated components for assessing scalability is provided below:

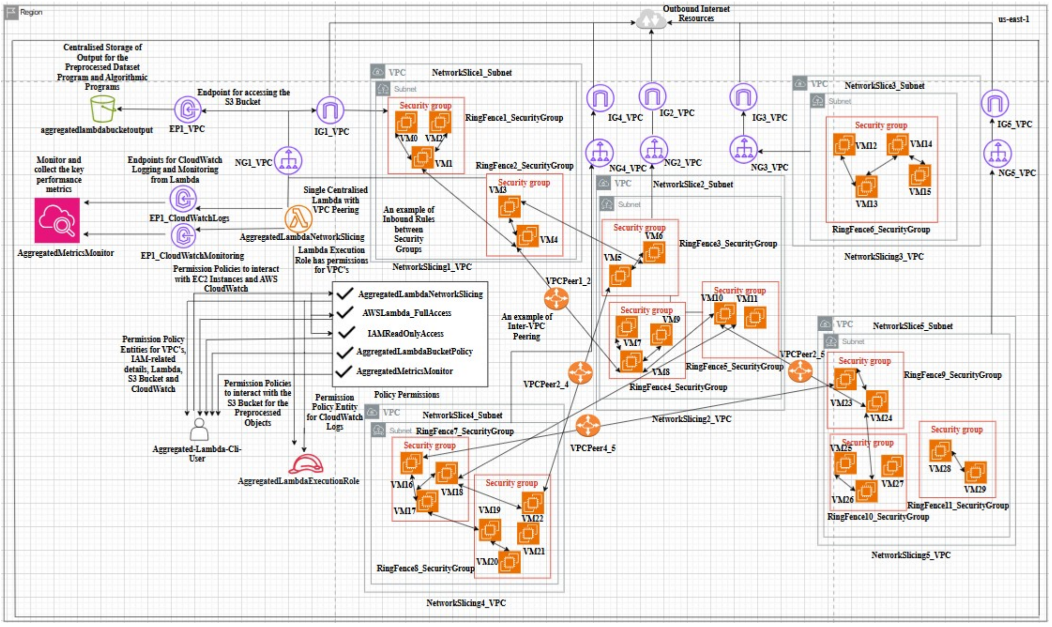

Region
Outbound Internet Resources
us-east-1
Centralised Storage of Output for the Preprocessed Dataset Program and Algorithmic Programs
aggregatedlambdabucketoutput
Endpoint for accessing the S3 Bucket
EP1_VPC
IG1_VPC
NG1_VPC
Monitor and collect the key performance metrics
AggregatedMetricsMonitor
Endpoints for CloudWatch Logging and Monitoring from Lambda
EP1_CloudWatchLogs
EP1_CloudWatchMonitoring
Single Centralised Lambda with VPC Peering
AggregatedLambdaNetworkSlicing
Lambda Execution Role has permissions for VPC's
Permission Policies to interact with EC2 Instances and AWS CloudWatch
✓ AggregatedLambdaNetworkSlicing
✓ AWSLambda_FullAccess
✓ IAMReadOnlyAccess
✓ AggregatedLambdaBucketPolicy
✓ AggregatedMetricsMonitor
Policy Permissions
Permission Policy Entities for VPC's, IAM-related details, Lambda, S3 Bucket and CloudWatch
Permission Policies to interact with the S3 Bucket for the Preprocessed Objects
Permission Policy Entity for CloudWatch Logs
Aggregated-Lambda-Cli-User
AggregatedLambdaExecutionRole
VPC
Subnet
NetworkSlice1_Subnet
Security group
RingFence1_SecurityGroup
VM0
VM2
VM1
RingFence2_SecurityGroup
VM3
VM4
An example of Inbound Rules between Security Groups
NetworkSlicing1_VPC
VPCPeer1_2
An example of Inter-VPC Peering
VPCPeer2_4
VPCPeer2_5
VPCPeer4_5
IG4_VPC
IG2_VPC
IG3_VPC
IG5_VPC
NG4_VPC
NG2_VPC
NG3_VPC
NG5_VPC
NetworkSlice2_Subnet
VM5
VM6
RingFence3_SecurityGroup
VM7
VM8
VM9
RingFence4_SecurityGroup
VM10
VM11
RingFence5_SecurityGroup
NetworkSlicing2_VPC
NetworkSlice3_Subnet
VM12
VM13
VM14
VM15
RingFence6_SecurityGroup
NetworkSlicing3_VPC
NetworkSlice4_Subnet
RingFence7_SecurityGroup
VM16
VM17
VM18
VM19
VM20
VM21
VM22
RingFence8_SecurityGroup
NetworkSlicing4_VPC
NetworkSlice5_Subnet
RingFence9_SecurityGroup
VM23
VM24
VM25
VM26
VM27
RingFence10_SecurityGroup
VM28
VM29
RingFence11_SecurityGroup
NetworkSlicing5_VPC

Figure 23: AWS Setup for the Aggregated Uneven Network Slicing Ring Fencing Architecture with respective requirements in us-east-1.

**Environmental Configuration**: The AWS environment above setup five VPC's with corresponding subnets consisting of eleven security groups that contained thirty VM's or EC2 instances. VPC's with corresponding subnets, security groups and VM's can be compared to network slices, ring fences and computational network resources. As an additional factor in uneven aggregation, the subnets were in different availability zones but defined in the same region. VPC peering setup inter-slice communication where corresponding route tables with explicit subnet association for those subnets were configured with outbound rules for security groups to allow only required traffic between VM's.
Likewise, Inter-Ring Fence communication was defined with inbound rules for security groups through VM-level access. To follow up on inter VM-VM connectivity between the respective security groups, each respective subnet ensured it's own NAT gateway and Internet Gateway for access to outbound internet resources. Next, the lambda function 'AggregatedLambdaNetworkSlicing' was utilised as a test event evaluating each algorithmic program against each other specifically for the last iteration. 'AggregatedLambdaNetworkSlicing' Lambda was associated with the first network slice titled 'NetworkSlicing1_VPC' to ensure peering connections to all other VPC's. After the test event is successfully run, the dataset based metrics such as CPU, memory, disk usage, network bandwidth are logged and monitored by 'AggregatedMetricsMonitor' CloudWatch.
'AggregatedMetricsMonitor' CloudWatch with it's two associated VPC endpoints facilitated CloudWatch Logs and CloudWatch Monitoring with collected performance metrics respectively. The logged performance metrics specific to 'AggregatedLambdaNetworkSlicing' function execution were viewed in custom dashboards to benchmark each algorithmic program on the basis of energy consumption, iteration time and task completion time. A more detailed view of the performance metrics specific to each run algorithmic program such as CloudSim metrics can be downloaded from the 'aggregatedlambdabucket' S3 Bucket as simulated output files. The AggregatedLambdaNetworkSlicing' test event that ran the algorithmic programs collected from 'aggregatedlambdabucket' was associated with a NAT gateway and attached VPC endpoint that routed private network resource traffic in their own private VPC's.

**IAM-specific permissions:** 'AggregatedLambdaBucketPolicy' and 'AggregatedMetricsMonitor' were attached policy permissions on 'AggregatedLambdaExecutionRole' that stored the output of the algorithmic programs with object keys that facilitated CloudWatch Logs and Monitoring respectively. These objects keys were the cleaned datasets preprocessed_fastStorage.csv and preprocessed_rnd.csv. 'AggregatedLambdaExecutionRole' was an IAM role that leveraged VPC peering for the remaining VPC's and IAM-related details for 'AggregatedLambdaNetworkSlicing' Lambda, 'aggregatedlambdabucketoutput' S3 bucket and 'AggregatedMetricsMonitor' CloudWatch. Attached to 'AggregatedLambdaExecutionRole' was a policy permission titled 'AggregatedLambdaNetworkSlicing' that fulfilled IAM-specific tasks to the Network Slicing Ring Fencing architecture. An IAM user titled 'Aggregated-Lambda-Cli-User' ensured correct association with all the aforementioned policy permissions including the 'AggregatedLambdaNetworkSlicing' policy execution role for all sources. For each incorporated program in AggregatedLambdaNetworkSlicing Lambda, the updated function code and IAM-specific details to the AggregatedLambdaNetworkSlicing policy were reflected in AWSLambda_FullAccess and IAMReadOnlyAccess for Aggregated-Lambda-Cli-User. AggregatedMetricsMonitor CloudWatch collected detailed performance metrics that was stored through the 'aggregatedlambdabucketoutput' S3 Bucket as a means of centralised storage for the programs.